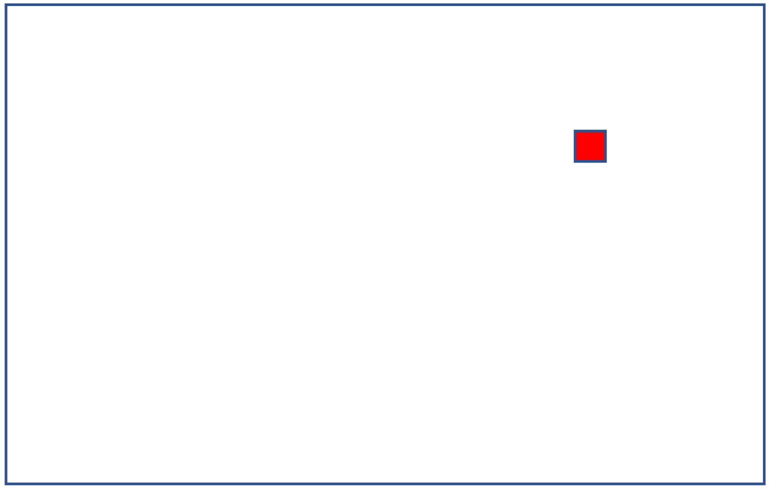
CMSC427
Drawing on the CPU vs. GPU

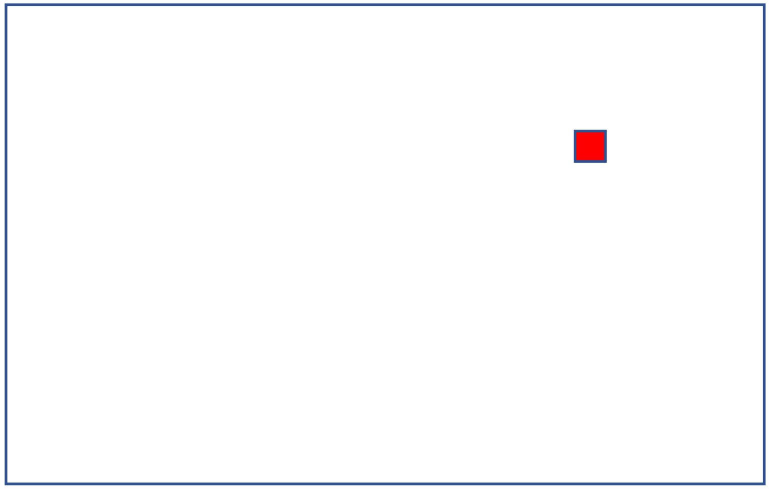
Basics of drawing: putting a pixel

PUTPIXEL(X,Y,R,G,B,A)



Basics of drawing: putting a pixel

PUTPIXEL(X,Y,R,G,B,A)



How much data to transfer?

Position X,Y – 16 bits each
Color – 8 bits each
- Red
- Green
- Blue
- Alpha (transparency)

Total: 8 bytes x # of pixels

Retaining state

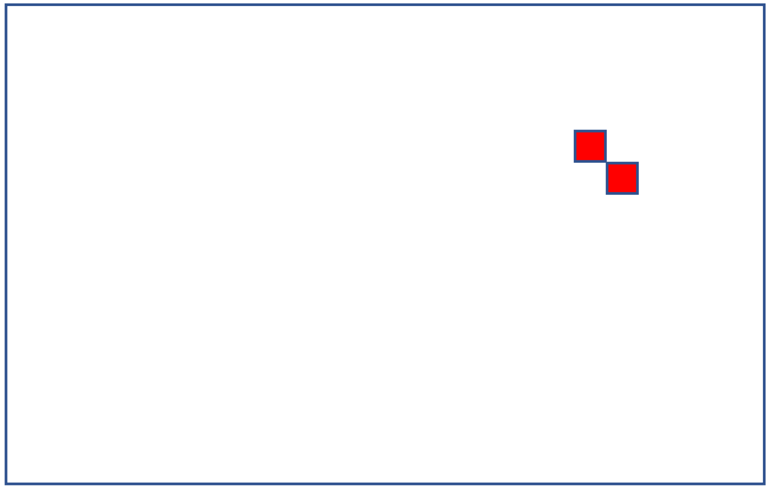
SETCOLOR(R,G,B,A)
PUTPIXEL(X,Y)
PUTPIXEL(X,Y)

SETCOLOR() sets ***state*** of graphics card, doesn't draw

PUTPIXEL() draws

Transfer color bits less often

More we can ”preload” on graphics card, less we need to do on CPU and then transfer



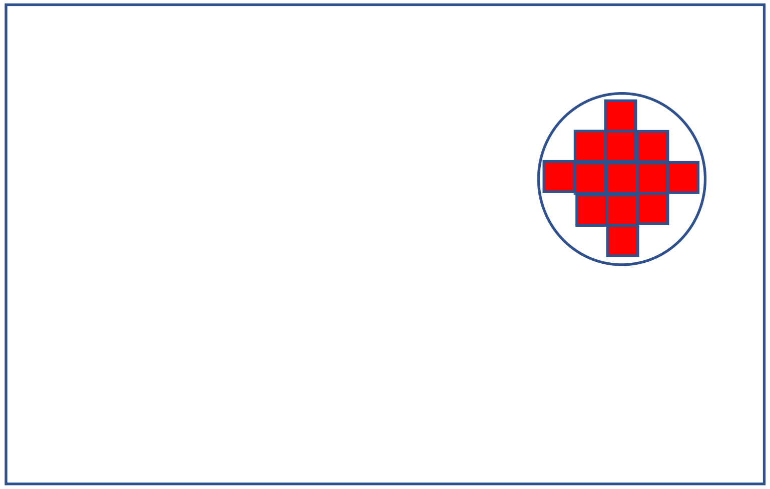
Delegating computation

SETCOLOR(R,G,B,A)
DRAWCIRCLE(X,Y,R)

SETCOLOR() sets state of graphics card, doesn't draw

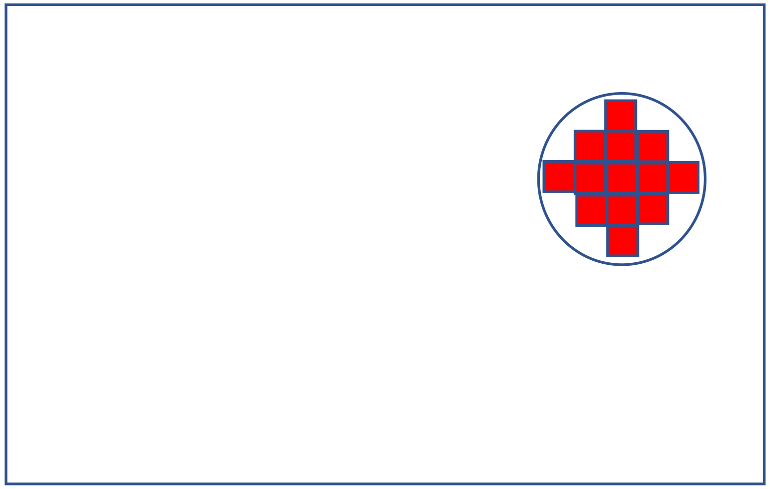
DRAWCIRCLE() draws by invoking routine on graphics card

Less data transferred. Transfer only values x,y,R but get all pixels in circle filled



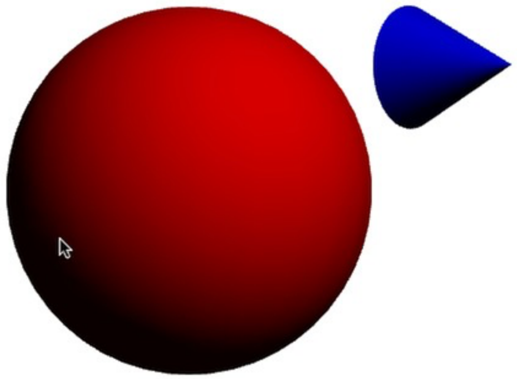
2D vs. 3D GPUs

DRAWCIRCLE(X,Y,R)

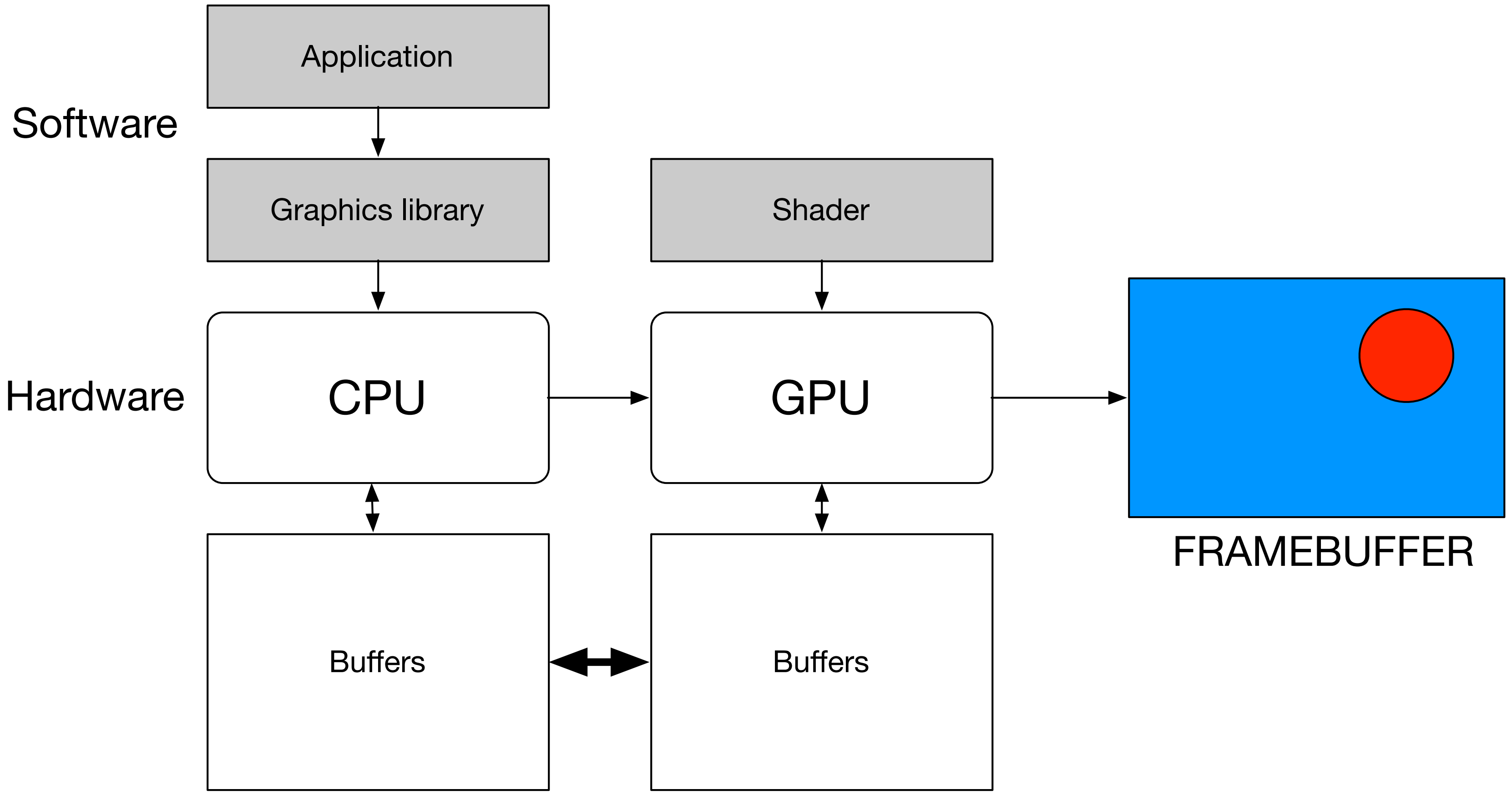


Simple 2D GPU – fixed primitives
Drawcircle, drawline, drawsquare
MoveRegion, CopyRegion
2D screen oriented

Advanced 3D GPU – programmable
Shaders allow programmers to set shape, lighting, other effects

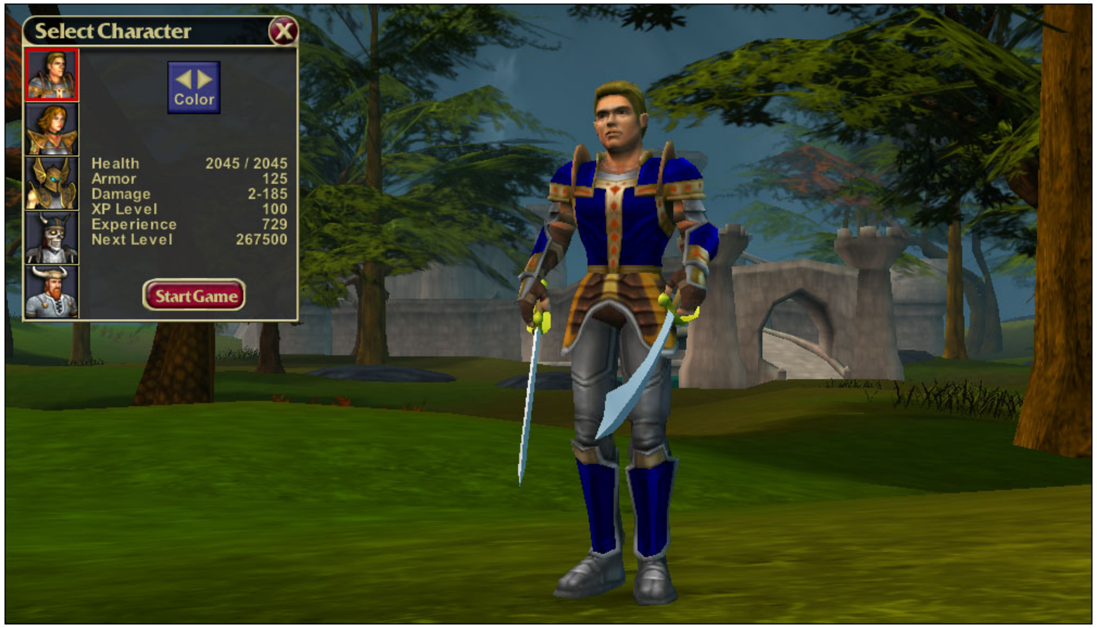


Drawing on CPU vs GPU



Preloading for online game

- Download to GPU in advance terrain model, character model, textures, character behaviors
- In gameplay only need to download changes in character state – movement, weapons, etc.
- Significantly reduce network bandwidth



Health	2045 / 2045
Armor	125
Damage	2-185
XP Level	100
Experience	729
Next Level	267500

What you should know

1. Where drawing happens – software rendering on CPU vs hardware rendering on GPU
2. Why we might preload data to the GPU
3. That GPUs have state that applies to objects drawn
4. That simple GPUs have fixed 2D primitives, and advanced GPUs have programmable 3D pipelines